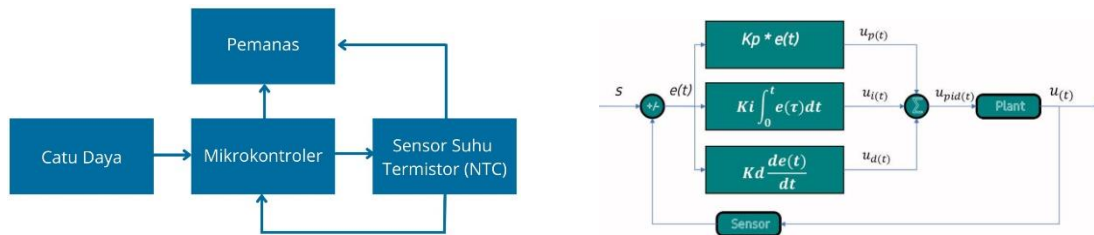


# LAPORAN PERCOBAAN

Aflaha Salsabila

22106020033

## 1. Perancangan (diagram Blok)



## 2. Pembuatan (Penjelasan cara Merakit)

- Alat dan bahan

No.	Alat dan Bahan	Fungsi
1.	Power Supply	Sebagai sumber daya untuk menjalankan komponen
2.	Arduino Uno	Sebagai pemrograman mikrokontroler dan pengendali otomatis
3.	Torlipy CR10 Blok Pemanas	Komponen pemanas yang dapat menghasilkan dan mempertahankan suhu
4.	Modul Driver Motor DC Daya Tinggi BTS7960	Sebagai pengendali pada pemanas
5.	Cartridge IJ Printing 3D	Mengatur suhu sehingga dapat mencapai suhu yang diinginkan
6.	Thermistor NTC100K	Mendeteksi perubahan suhu
7.	Stand Blok Pemanas	Sebagai tempat untuk blok pemanas
8.	Resistor 100K	Digunakan untuk pembagi tegangan
9.	Breadboard	Papan rangkaian tanpa perlu menyolder
10.	Kabel Jumper	Penghubung antar komponen

- Proses Pembuatan

Pada percobaan kali ini sebuah sistem pemanas yang dikendalikan oleh mikrokontroler Arduino Uno. Dimana pada sensor suhu berupa thermistor dipasang pada blok pemanas dan dihubungkan pada pin A0 dengan Arduino sebagai input pada pembacaan sensor, serta pastikan terpasang dengan baik untuk mendapatkan akurasi pengukuran. Kemudian sambungkan ke VCC pada Arduino dengan ditambahkan resistor 100K sebagai pembagi tegangan dan Ground pada Arduino. Lalu memasukkan Cartridge (semikonduktor) ke dalam blok pemanasnya untuk mengukur suhu di area pemanas.

Selanjutnya menghubungkan heater PWM ke pemanas sebagai pengontrol daya, dimana PWM (Pulse Width Modulation) adalah untuk mengatur daya dengan mengubah lebar pulsa sinyal. Menghubungkan pin heater PWM dari modul yang dihubungkan ke pin 9 pada Arduino, modul tersebut berfungsi sebagai perantara antara Arduino dan pemanas untuk mengatur daya yang lebih besar.

Langkah berikutnya adalah modul tersebut disambungkan pada VCC dan Ground ke Arduino. Modul driver motor DC digunakan untuk mengontrol daya yang diberikan ke pemanas. Pada pin LPWM (Low-Side PWM) pada modul dihubungkan ke pin 12 pada Arduino untuk memberikan sinyal PWM yang akan mengontrol daya ke pemanas.

Adapun pada elemen pemanas yaitu kabel pemanas yang akan menghasilkan panas Ketika dialiri arus listrik. Dimana dihubungkan dengan salah satu ujung kabel tersebut ke output modul. Dan untuk ujung lainnya dihubungkan ke power supply untuk memberikan daya pada pemanas.

Secara keseluruhan, rangkaian tersebut bekerja sebagai berikut yaitu, thermistor mengukur suhu di area pemanas dan mengirimkan sinyal pada Arduino. Lalu Arduino akan membaca nilai suhu dari thermistor dan membandingkan dengan nilai suhu yang diinginkan. Kemudian Arduino juga menghitung nilai PWM yang sesuai dan mengirimkannya ke modul BTS7960. Maka dari itu, modul dapat mengubah sinyal PWM menjadi sinyal listrik yang diberikan ke pemanas. Pada pemanas nantinya akan menghasilkan panas sesuai dengan besarnya daya yang diberikan.

Sehingga dari percobaan kali ini didapatkan beberapa tujuan yaitu, menguji sistem kontrol pemanas, menganalisis kerja sistem, dan dapat menentukan nilai pengontrol ( $K_p$ ,  $K_i$ ,  $K_d$ ).

3. Pengujian risetime dan overshoot sistem kendali pemanas dengan variasi  $K_p$ ,  $K_i$ ,  $K_d$  (jelaskan hasil pengujian)

<i>Percobaan</i>	<i>K<sub>p</sub></i>	<i>K<sub>i</sub></i>	<i>K<sub>d</sub></i>	<i>Current Temperature</i>	<i>PWM Output</i>	<i>Steadystate Error</i>
1	0	0	0	-50,30°C	0,00	50,30°C
2	1	0.01	0	-18,41°C	69,36	48,41°C
3	15	0.008	0.001	12,60 °C	255,0	17,40°C

Pada percobaan kali ini untuk mendapatkan hasil yang sesuai diperlukan mempersiapkan serta menyusun komponen yang telah dijelaskan diatas. Menginisialisasi program, pastikan bahwa program telah diunggah ke Arduino dengan variabel untuk parameter  $K_p$ ,  $K_i$ ,  $K_d$  yang bisa disesuaikan. Serta menetapkan suhu target atau *setpoint* pada program, pada percobaan ini menggunakan *setpoint* sebesar 30°C untuk menguji seberapa dekat sistem dapat mencapai suhu yang diinginkan.

Pada percobaan pertama yaitu dengan nilai  $K_p$  sebesar 0,  $K_i$  sebesar 0, dan  $K_d$  sebesar 0. Menyetel nilai tersebut pada program, lalu menyalakan sistem dan mencatat suhu yang telah tercapai oleh pemanasnya hingga mencapai keadaan stabil maka didapatkan hasil untuk *Current Temperature* sebesar -50,30°C, *PWM Output* sebesar 0,00 dan *Steadystate Error* sebesar 50,30°C. Hasil tersebut menunjukkan bahwa tanpa kontrol, sistem tidak dapat mencapai suhu yang jauh lebih rendah dari *setpoint*.

Percobaan kedua yaitu dengan mengubah nilai  $K_p$  sebesar 1,  $K_i$  sebesar 0,01 dan  $K_d$  sebesar 0. Menyalakan sistem kembali dengan memasukkan nilai tersebut dan diamati bagaimana perubahan suhu dan stabilitasnya serta mencatat suhu yang telah tercapai yaitu untuk *Current Temperature* sebesar -18,41°C, *PWM Output* sebesar 69,36 dan *Steadystate Error* sebesar 48,41°C. Hasil tersebut menunjukkan bahwa dengan peningkatan kecil pada  $K_p$  dan  $K_i$ , sistem

mulai merespons lebih baik terhadap *setpoint*, meskipun mungkin masih terdapat eror yang cukup besar.

Percobaan terakhir yaitu ketiga dengan mengubah nilainya untuk  $K_p$  sebesar 15,  $K_i$  sebesar 0,008 dan  $K_d$  sebesar 0,001. Menyalakan sistem kembali dengan memasukkan nilai tersebut dan diamati bagaimana perubahan suhu dan stabilitasnya serta mencatat suhu yang telah tercapai yaitu untuk *Current Temperature* sebesar 12,60 °C, *PWM Output* sebesar 255,0 dan *Steadystate Error* sebesar 17,40°C. Dari hasil tersebut nilai pada PWM menjadi lebih tinggi dan suhu akan mendekati *setpoint* lebih baik dibandingkan dengan percobaan sebelumnya.

Maka dari itu, dapat dianalisis dan disimpulkan bahwa hasil dari ketiga percobaan, terutama pada *Current Temperature*, *PWM Output*, dan *Steadystate Error*. Dapat diidentifikasi dengan peningkatan parameternya yang memberikan pengaruh signifikan pada sistem untuk mencapai suhu yang diinginkan.

Dari pecobaan tersebut masih terjadi beberapa kesalahan sehingga dapat dilakukan penyempurnaan jika diperlukan. Dengan cara jika hasil masih kurang optimal, pertimbangkan meode tuning PID lainnya dan melakukan penyesuaian hingga sistem mencapai hasil yang stabil dan sesuai.